AFB
IFW

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Inventor(s): **Sarvar Patel**

Case: **18**

Serial No.: **09/854251** Group Art Unit: **2137**

Filing Date: **May 11, 2001**

Examiner: **C. D. Fields**

Title: **Message Authentication System And Method**

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

Enclosed in triplicate is an Appeal Brief in the above-identified patent application.

Please charge the amount of **$500** covering payment of the fee for the Appeal Brief to **Lucent Technologies Inc. Deposit Account 12-2325**. Triplicate copies of this letter are enclosed.

In the event of non-payment or improper payment of a required fee, the Assistant Commissioner is authorized to charge or to credit **Deposit Account No. 12-2325** as required to correct the error.

Respectfully,

Martin I. Finston
Attorney for the Applicant
Reg. No. 31613
(973)-386-3147

Date: March 16, 2006

Docket Administrator (Room 3J-219)
Lucent Technologies Inc.
101 Crawfords Corner Road
Holmdel, NJ 07733-3030

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop **APPEAL BRIEF-PATENTS**, Director of the US Patent and Trademark Office, PO Box 1450, Alexandria, VA 22313-1450, on March 17, 2006.

Margaret Cardoso
Margaret Cardoso

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

Patent Application
Inventor(s): **Sarvar Patel**

Case: **18**

Serial No.: **09/854251** Group Art Unit: **2137**

Filing Date: **May 11, 2001**

Examiner: **C. Fields**

Title: **Message Authentication System And Method**

Mail Stop Appeal Brief – Patent
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

Appellant submits an original and two copies of a Brief on Appeal as required by 37 C.F.R. § 41.37.

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **MS Appeal Brief-Patents**, Commissioner for Patents, Alexandria, VA 22313-1450 on March 17, 2006.
Margaret Cardoso
Margaret Cardoso

TABLE OF CONTENTS

BRIEF ON BEHALF OF APPELLANT

In support of the Notice of Appeal filed on January 19, 2006, appealing the Examiner's final rejection mailed on September 20, 2005 of each of pending claims 1-4, 7-11, 14, 16, and 19-21 of the present application which appear in the attached claims appendix, Appellant hereby provides the following remarks.

I. REAL PARTY IN INTEREST

The real party in interest in this appeal is Lucent Technologies Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences that will affect, be directly affected by, or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

Claims 1-4, 7-11, 14, 16, and 19-21 are pending in the application, with claims 1, 7, 14, 16, 19, and 21 being independent.

Claims 1-4, 7-11, 14, 16, and 19-21 are finally rejected under 35 U.S.C. §102(b) as being unpatentable over Bellare.

Claims 1-4, 7-11, 14, 16, and 19-21 are being appealed.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed invention is directed to methods and apparatus for using hash functions to process a message for authentication.

As provided in claim 1, different processing is invoked, depending on whether the message fits within an input block of a compression function. (Specification, page 10, line 29, to page 12, line 2.

If the message fits within the said block, a single iteration of the compression is carried out. The inputs for said single iteration are: the message, and a key. A result from the compression function, without further iteration thereof, is used to produce a message authentication code (MAC). Such a procedure is illustrated in FIG. 9 of the instant specification, where element 90 represents the compression function.

If the message does not fit within the said block, the message is processed using a hash function nested within a keyed hash function. Then, a result from the keyed hash function is used to produce the MAC. Such a procedure is illustrated in FIG. 10, where the inner hash function F_{CV2} (represented in the figure by elements 100a, . . . , 100n) is nested within outer hash function F_{CV1} (represented in the figure by element 102. As indicated in the figure, one of the inputs to F_{CV1} is the key CV1.

In the specific example illustrated by FIG. 10, the message, denominated "X", is divided into portions denominated "PORTION 1" and "PORTION 2". As shown in the figure, PORTION 1 fits within an input block of hash function 102, but PORTION 2 may need to be further subdivided in order to be processed by a hash function.

It will be seen in FIG. 10 that the notation "f_{CV1}(PORTION 1, F_{CV2}(PORTION 2))" is used to indicate the output of hash function 102. Such notation makes it clear that the "inner" function F_{CV2} is nested within the "outer" function f_{CV1}.

Claim 1 is directed to a method. Claim 14 is directed to the corresponding apparatus. Hence, the above comments apply *mutatis mutandis* to claim 14.

As provided in claim 7, the message has a first portion which is processed as input to a hash function. The message also has a second portion, which is concatenated with the result of said hash function. The concatenation is then processed as input to a keyed hash function.

Reference to FIG. 10 will show that the input to hash function 102 is an example of such a concatenation. In the example of FIG. 10, PORTION 2 is processed as input to the hash function represented by elements 100a, . . . , 100n. The result is concatenated with PORTION 1 and hashed by element 102, keyed with CV1.

Claim 7 is directed to a method. Claim 16 is the corresponding apparatus claim. Hence, the above comments apply *mutatis mutandis* to claim 16.

Claim 19 is directed to a method in which the message, or a portion thereof, is compressed by a hash function on condition that the message exceeds a block size. Then either the message or an intermediate result due to the hash function, is compressed by a keyed compression function, and the result is used in a message authentication scheme. FIG. 10 illustrates an example of such a method, in which the "intermediate result" is the concatenation of F_{CV2}(PORTION 2) with PORTION 1.

Claim 19 is directed to a method. Claim 21 is the corresponding apparatus claim. Hence, the above comments apply *mutatis mutandis* to claim 21.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant seeks the Board's review of the rejection of Claims 1-4, 7-11, 14, 16, and 19-21 under 35 U.S.C. §102(b) as unpatentable over Bellare.

VII. ARGUMENTS

A. Claims 1-4 and 14 are not anticipated by Bellare.

As claims 1-4 and 14 are allowable for features which are present in each claim, the arguments below are directed to claim 1, with claim 1, its dependent claims 2-4, and independent claim 14, all standing or falling together.

Bellare has been cited as disclosing certain hash functions and MAC functions.

Specifically, the Examiner cited Bellare, page 8, as disclosing the keyed hash function $F_k(x_1, x_2, \ldots, x_n)$, which is diagrammed at Specification, FIG. 6. The

Examiner cited Bellare, page 9, as disclosing the MAC function $NMAC_k(x)$, where $x = (x_1, x_2, \ldots, x_n)$. $NMAC_k(x)$ is diagrammed at Specification, FIG. 7. The Examiner cited Bellare, page 13, as disclosing the MAC function $HMAC_k(x)$, which is diagrammed at Specification, FIG. 8.

In regard to $F_k(x_1, x_2, \ldots, x_n)$, it will be seen that if the input message x is longer than one block length, the hash function will be carried out by iterating the compression function f. Significantly, the key k is applied only to the initial iteration. Taken as a whole, $F_k(x_1, x_2, \ldots, x_n)$ is a "keyed hash function" because the key k is one of the inputs, and the output is a hashed version of $(x_1, x_2, \ldots, x_n)$.

However, with reference to the language of claim 1, $F_k(x_1, x_2, \ldots, x_n)$ fails to use "a hash function nested within a keyed hash function". That is, the quoted language in the preceding sentence describes an inner function ("a hash function") nested within an outer function ("a keyed hash function"). Significantly, the quoted language specifies that the outer, and not the inner, function must be keyed.

With reference to FIG. 6 of the instant Specification, attention is drawn to the fact that each instance of the compression function f is nested within the function or functions that follow it. The key K (as seen in the figure) is applied as input only to the first instance of the compression function. Thus, the keyed hash function can only be an inner function, and never an outer function.

Accordingly, $F_k(x_1, x_2, \ldots, x_n)$ achieves the opposite of what claim 1 describes. Instead of providing "a hash function nested within a keyed hash function", $F_k(x_1, x_2, \ldots, x_n)$ provides a keyed hash function which itself is the nested function.

In regard to $NMAC_k(x)$, it will be seen that even if the input message x fits within a single block length, the compression function f must be called at least twice. The first instance of f, as shown at Specification, FIG. 7, is the instance that takes k_2 as an input key, and the last instance is that which takes k_1 as an input key. The first and last instances must both take place, even for the shortest input messages.

Thus, with reference to the language of claim 1, when the message fits within one input block, $NMAC_k(x)$ fails to satisfy the requirement of "performing a single iteration of the compression function using a key and said message as inputs . . . and using a result from said compression function *without further iteration thereof* to produce a message authentication code". [Emphasis added.]

In regard to $HMAC_k(x)$, it will be seen from Specification, FIG. 8, that even if the input message x fits within a single block length, the compression function f must be called at least four times: twice taking initial variable IV and a padded key k as input, at least once taking at least a portion of the message as input, and once taking the hashed message and a key derived from k and IV as input. Thus, $HMAC_k(x)$ fails to meet the limitations of claim 1 for the same reasons set out above in regard to $NMAC_k(x)$.

B. Claims 7-11 and 16 are not anticipated by Bellare.

As claims 7-11 and 16 are allowable for features which are present in each claim, the arguments below are directed to claim 7, with claim 7, its dependent claims 8-11, and independent claim 16, all standing or falling together.

Claim 7 is drawn to a method of processing an input message, in which a "first portion" of the input message is hashed. A second portion of the input message is combined with the hashed first portion, and the combination is used as input to a keyed hash function. The combination is formed by concatenating the second portion with the hashed first portion.

Bellare fails to disclose any hash function or MAC-generating function in which two portions of an input message are processed differently, and then concatenated, and the concatenation used as input for further processing.

Turning to Specification, Figures 6-8, it will be seen that all of the processing stages (i.e., instances of the compression function f) take two inputs. However, none of these inputs is a concatenation of two differently processed portions of a message. If two differently processed portions of a message are to be fed into a processing

stage of $F_k(x_1, x_2, \ldots, x_n)$, $NMAC_k(x)$, or $HMAC_k(x)$, one portion will be input as the data block (the upper input x_i in the figures), and the other portion will be input as the chaining variable (the lower input as seen in the figures). The portions will not be concatenated together and presented as a single input.

C. Claims 19-21 are not anticipated by Bellare.

As claims 19-21 are allowable for features which are present in each claim, the arguments below are directed to claim 19, with claim 19, its dependent claim 20, and independent claim 21, all standing or falling together.

According to the method of claim 19, there must be a conditional step of using a hash function to compress at least a portion of x, on condition that x exceeds a block size. The output of the conditional step is an intermediate result. The input message x or the intermediate result is then compressed using a keyed compression function.

Attention is directed first to the function $F_k(x_1, x_2, \ldots, x_n)$. Reference to FIG. 6 of the instant Specification will show that if the message is longer than one block size, the first and every intermediate processing stage of $F_k(x_1, x_2, \ldots, x_n)$ produces *arguendo* an "intermediate result". However, not a single one of those "intermediate results" is compressed with a keyed compression function as required by claim 19. Instead, at each of those intermediate stages, one block of the message is input as the data block, and the output of the previous stage is input as the chaining variable. No key is input to any of the intermediate stages, or to the final stage.

Attention is next directed to $NMAC_k(x)$ and $HMAC_k(x)$. Reference to FIGs. 7 and 8 of the Specification will show that both of said functions process the message and *arguendo* produce "intermediate results" as described above in reference to $F_k(x_1, x_2, \ldots, x_n)$. Moreover, the last stage of $NMAC_k(x)$ and likewise the last stage of $HMAC_k(x)$ *arguendo* takes an "intermediate result" as input and compresses it with a keyed compression function.

However, there is nothing conditional about processing the message to provide an intermediate result. Instead, as will be clear from the figures, even if the

message fits within a single block size, both $NMAC_k(x)$ and $HMAC_k(x)$ must take an intermediate result and, in the final processing stage, compress it with a keyed compression function.

Thus, "processing the message to provide an intermediate result" is not conditional, and therefore the limitations of claim 19 are not met.

VIII. EVIDENCE AND RELATED APPEALS APPENDICES:

As there are no related appeals and interferences, copies of decisions rendered by a court or the Board for such proceedings do not exist and have not been supplied in an Appendix pursuant to 41.37(c)(1)(x).

As no evidence was submitted and relied upon in this Appeal, an Appendix pursuant to 41.37(c)(1)(ix) has not been supplied.

IX. CONCLUSION

Appellant respectfully requests the Board to reverse the Examiner's anticipation rejection of claims 1-4, 7-11, 14, 16, and 19-21.

Respectfully submitted,

Sarvar Patel

By _____

Martin I. Finston, Attorney
Reg. No. 31613
908-582-2908.

Date: March 16, 2006

Docket Administrator (Room 3J-219)
Lucent Technologies Inc.
101 Crawfords Corner Road
Holmdel, NJ 07733-3030

Attachment: Appendix X

X. CLAIMS APPENDIX

1. (Currently Amended) A method of processing a message for authentication, said method comprising:

determining whether said message fits within an input block of a compression function;

performing a single iteration of the compression function using a key and said message as inputs when said message fits within an input block of said compression function and using a result from said compression function without further iteration thereof to produce a message authentication code; and

using a hash function nested within a keyed hash function to process said message when said message does not fit within an input block of said compression functionand using a result from said keyed hash function to produce a message authentication code.

2. (Original) The method of claim **1** wherein said step of using comprises the steps of:

providing a first portion and a second portion of said message;

performing a hash function using said first portion as an input to achieve a result; and

performing a keyed hash function using said second portion and said result as inputs.

3. (Original) The method of claim **2** wherein said hash function is an iterated hash function F and said keyed hash function is a keyed compression function f.

4. (Original) The method of claim **2** wherein said hash function is an iterated hash function F and said keyed hash function is an iterated hash function F.

5-6. (Cancelled)

7. (Currently Amended) A method of processing a message for authentication, said method comprising:

providing a first portion and a second portion of said message;

performing a hash function using said first portion as an input to achieve a result; and

performing a keyed hash function using a concatenation of said second portion and said result as input.

8. (Original) The method of claim **7** comprising the step of:

determining whether said message fits within an input block of a compression function; and

performing said steps of providing, performing and performing when said message does not fit within an input block of said compression function.

9. (Original) The method of claim **7** comprising the step of:

determining whether said message fits within an input block of a compression function; and

performing a single iteration of a compression function using a key and said message as inputs when said message fits within an input block of said compression function.

10. (Original) The method of claim **7** wherein said hash function is an iterated hash function F and said keyed hash function is a keyed compression function f.

11. (Original) The method of claim **7** wherein said hash function is an iterated hash function F and said keyed hash function is an iterated hash function F.

12-13. (Cancelled)

14. (Currently Amended) A message authentication system comprising:

processing circuitry configured to determine whether a message fits within an input block of a compression function; and

processing circuitry configured to perform a compression function using a key and said message as inputs, and to output a message authentication code after a single iteration of the compression function in the event that the message fits within one said block, but to use a hash function nested within a keyed hash function to process said message when said message does not fit within one said block.

15. (Cancelled)

16. (Currently Amended) A message authentication system comprising:

processing circuitry configured to provide a first portion and a second portion of a message, perform a hash function using said first portion as an input to achieve a result, and perform a keyed hash function using a concatenation of said second portion and said result as input.

17-18. (Cancelled)

19. (New) A method of processing a message x for authentication, comprising:

(a) conditionally processing x to provide an intermediate result y;

(b) compressing x or y with a keyed compression function having a block size; and

(c) providing a result of the compressing step for use in a message authentication scheme,

wherein (a) comprises using a hash function to compress at least a portion of x and is carried out on condition that x exceeds the block size.

20. (New) The method of claim **19** wherein (a) comprises providing a first portion and a second portion of the message x, performing a hash function using the first portion as an input to achieve a result; and concatenating the result with the second portion.

21. (New) A message authentication system comprising:

processing circuitry configured to determine whether a message x is larger than an input block size b of a keyed compression function;

processing circuitry configured to apply a hash function to compress at least a portion of x, thereby to provide an intermediate result y, said processing circuitry to be activated only in the event that x is larger than b; and

processing circuitry configured to compress x or y with said keyed compression function, thereby to provide a result for use in a message authentication scheme.